Towards Cooperative Automated Driving: Geographic-Aware Network Analysis and Visualization tool

Koichi Kambara
The University of Tokyo
kambara@hongo.wide.ad.jp

Ehsan Javanmardi
The University of Tokyo
ejavanmardi@g.ecc.u-tokyo.ac.jp

Jin Nakazato
The University of Tokyo
jin-nakazato@g.ecc.u-tokyo.ac.jp

Yousuke Watanabe
Nagoya University
watanabe@coi.nagoya-u.ac.jp

Kenya Sato
Doshisya University
ksato@mail.doshisya.ac.jp

Hiroaki Takada
Nagoya University
hiro@ertl.jp

Manabu Tsukada
The University of Tokyo
tsukada@hongo.wide.ad.jp

ABSTRACT

In recent years the cooperative automated vehicle (CAV) concept has been gaining attention due to its potential to increase traffic safety and traffic flow by utilizing the vehicle-to-everything communication capability. One of the key requirements for CAV is ensuring every vehicle receives relevant messages at the right time and place; therefore, measuring and visualizing network performance is vital. However, for CAV application, more network analyzers than those extant are needed because these do not consider geographical characteristics. In this study, we proposed a geographically-aware CAV-specific network analysis and visualization tool that can report the network performance factors such as packet loss, bandwidth, and jitter in real time. Further, we developed a proposal tool and evaluated it in an outdoor proof-of-concept study at the University of Tokyo's Hongo Campus.

KEYWORDS

Cooperative ITS, V2X, Dynamic Map, Network Analyzer

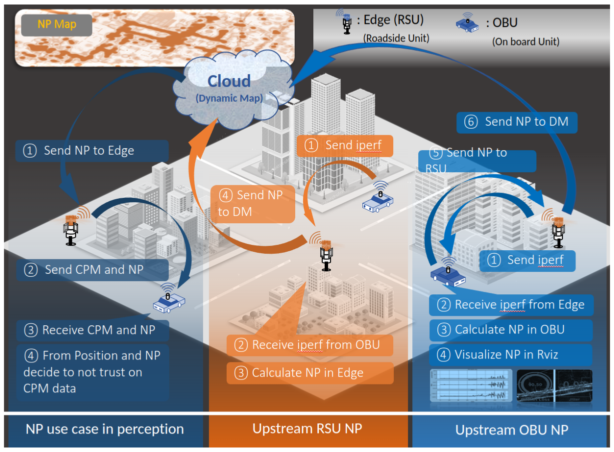

Figure 1: Overview of the proposed NP Analyzer and Visualizer

1 INTRODUCTION

In addition, to stand-alone perception sensors such as LiDARs and cameras in vehicles, through cooperative automated vehicle (CAV), other vehicles and infrastructure (roadside units (RSU)) can share their perceptions using cooperative awareness messaging (CAM) [1] and use cooperative perception messages (CPM) [2, 3] to see perceive objects beyond sight to improve safety. However, because it is challenging to guarantee communication reliability due to signals being blocked by obstacles, the distances to transmitters, and the instability of the wireless equipment itself, ensuring that each vehicle receives the message at the right time and in the right place is difficult. Therefore, constant network performance (NP) analysis is essential for the vehicles to decide which perception mode to adopt. Moreover, their positions should be considered as the NP is highly related to the geographic positioning of the vehicle and RSU. However, existing network analyzer such as iperf and ping does not have this feature. In contrast, AnaVANET [4] is a Vehicle-to-Everything (V2X) communication analysis tool that considers geographic characteristics. This tool can analyze GeoNetworking, a protocol for V2X communication, and is used for post-processing test data and visualizing NP when field tests are performed. However, the problem with this tool is that it processes test data after all data have been collected, thus losing real-time performance. On the other hand, visualizing the network analysis results can help researchers better investigate the sources of system misoperations. In this work, we introduce a real-time NP analysis and visualization tool that can report and visualize the packet loss of the communication links, bandwidth, and jitter for CAV applications as shown in Figure. 1. The calculated performance factors can be integrated into the map so that vehicles can decide whether to rely on collective perception.

AINTEC 2022, Dec 19–21, 2022, Hiroshima, Japan
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 NETWORK ANALYZER AND VISUALIZER

2.1 Requirements

For AnaVANET, create a visualization tool for cooperative automated driving that meets the following specifications:

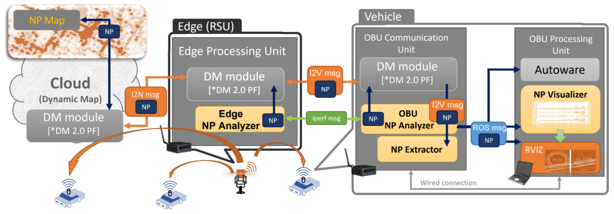

Figure 2: Integration architecture of our NP analyzer and visualizer with DM2.0PF

- Communication paths and performance per link
- Geographical awareness
- Adaptation to various scenarios
- Ease of data collection
- Real time operation

2.2 System Architecture

Figure 2 describes the system architecture. During system operation, a fixed number of packets per unit of time are multicast by iperf from the Edge to the vehicle and anycast from the vehicle to the Edge. The packets flowing through iperf are captured by each NP Analyzer (ROS node) using Wireshark to calculate a summary of the packet loss, bandwidth, and delay per second. The communication status from the Edge to the vehicle is published as a ROS message and visualized on Autoware [5] by subscribing to Netowork visualization. In addition, the communication status is converted into stream data that can be handled by DM2.0PF [6] and uploaded to the database. The stream data flowing to the database is anycast to the Edge and multicast from the Edge to other vehicles using the communication function of DM2.0PF. The stream data of other vehicles received through DM2.0PF is converted into ROS messages by NP Extractor. The published data is subscribed by NP Visualizer to visualize the communication status of other vehicles using matplotlib.

2.3 Implementation

This paper proposes measuring and visualizing packet loss, bandwidth, and jitter. NP analysis is performed by the function of NP Analyzer. When measuring real-time communication data, iperf is used to fix the number of packets sent per unit of time because it is difficult to measure packet loss on the receiving side without knowing the number of packets sent. Besides, since packet loss per link during multi-hopping is unknown, packets sent by iperf are captured using Wireshark. We then measure the packet loss per link from the obtained tcpdump. On the other hand, bandwidth is calculated from the size of packets captured by Wireshark, and jitter is calculated using a method based on a formula defined in RFC3550 [7].

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \tag{1}$$

$$J(i) = J(i-1) + (|D(i-1,1)| - J(i-1))/16 \tag{2}$$

S is the packet timestamp and R is the packet arrival time, D represents the difference in packet interval between the sender and receiver, and J is the jitter. /par

The measured communication status was visualized using RViz and matplotlib, standard visualization tools of Autoware, an Open Source Software (OSS) for automated driving, for versatility. First, in RViz, multicast packet loss from the edge to the vehicle was shown as a pie graph with numerical values, and jitter was visualized on the overlay as a line graph, respectively. The color of the graph changes according to the numerical value, allowing the user to determine whether the condition is normal or abnormal. In addition, more detailed communication status (packet loss, bandwidth, and jitter) is visualized on matplotlib line graphs.

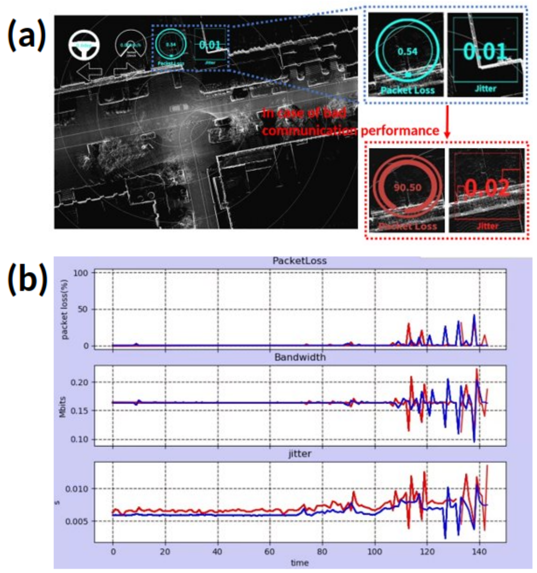

Figure 3: Visualizing NP (a) Visualizing real time NP in RViz. (b) Report NP in line graphs (Top: Packet loss, Middle: Bandwidth, Bottom: Jitter

Data, including communication status, is shared with all vehicles and edges in the same network using DM2.0PF [8]. For the integration of Autoware and DM2.0PF, we converted the ROS messages handled by Autoware and the stream data handled by DM2.0PF. The NP obtained by NP Analyzer is uploaded to DM2.0PF and downloaded from DM2.0PF by NP Extractor. By monitoring the communication status in DM2.0PF, data can be easily collected in the cloud.

2.4 Proof of Concept Results

We evaluated proposal tool in an outdoor proof-of-concept (PoC) at the Hongo Campus, the University of Tokyo, Japan. At first, the communication status was measured with the edge installed at a different position while moving a cart equipped with Lidar. Second, we visualized how much communication was possible at that location by moving the cart while estimating its self-positioning with Lidar and Autoware on the cart. Figure 3(a) shows the visualization using RViz, which displays packet loss and jitter as a graph, with the colors changing according to the NP. In addition, the communication performance between the edge and the moving vehicle is displayed. Figure 3(b) shows a matplotlib visualization of packet loss, bandwidth, and jitter, with the multicast from the edge to the vehicle in blue and anycast from the vehicle to the edge in red. The horizontal axis is time, updated once a second in real time.

3 CONCLUSION

In this study, we created a network analysis and visualization tool for cooperative automated driving. We confirmed that the tool met the requirements specified above and operated as intended throughout the demonstration PoC.

REFERENCES

[1] Intelligent Transport Systems; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service, 2019. ETSI EN 302 637-2 V1.4.1 (2019-04).

[2] Intelligent Transport Systems; Cooperative Perception Services (CPS), December 2019. ETSI TS 103 324 V0.0.14 (2019-10).

[3] Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2 , December 2019. ETSI TR 103 562 V2.1.1 (2019-12).

[4] M. Tsukada et.al. AnaVANET: an experiment and visualization tool for vehicular networks. In 9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TRIDENTCOM 2014), Guangzhou, China, May 2014.

[5] S. Kato et.al. Autoware on board: Enabling autonomous vehicles with embedded systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems(ICCPS), pp. 287–296. IEEE, 2018.

[6] Dynamic Map 2.0 Consortium, http://www.nces.i.nagoya-u.ac.jp/dm2/

[7] RFC 3550 - RTP: A Transport Protocol for Real-Time Applications, July 2003.

[8] Y. Watanabe et.al. DynamicMap 2.0: A Traffic Data Management Platform Leveraging Clouds, Edges and Embedded Systems, 2020. International Journal of Intelligent Transportation Systems Research 18:77–89.